

Examen XML (Sujet B)

— Master d'Ingénierie Informatique —

Mars 2013, durée 2h.

Les réponses apportées aux questions doivent être justifiées avec clarté et concision. Les documents sont interdits à l'exception d'un dictionnaire de memento (4 pages A4). L'examen est composé de quatre exercices indépendants. Les appareils électroniques sont interdits à l'exception des montres. Les réponses aux questions doivent directement être écrites dans les cadres du sujet.

► Exercice 1 On considère les trois schémas XML suivants.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="head" type="xsd:integer"/>
        <xsd:element name="tail" type="xsd:integer" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="head" type="xsd:integer"/>
        <xsd:element name="tail" type="xsd:integer"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="head" type="xsd:integer"/>
  <xsd:element name="tail" type="xsd:integer" substitutionGroup="head"/>
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="head"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

a) Donner un document valide pour le premier schéma.

b) Donner un document valide pour le deuxième schéma mais non valide pour le premier schéma.

c) Donner un document valide pour le troisième schéma mais non valide pour les deux premiers.

d) Donner un document valide pour les trois schémas.

e) Pour chacun des trois schémas donner une DTD équivalente.

► Exercice 2 On considère la DTD ci-dessous.

```
<!ELEMENT tree ((tree,tree)?,val)>
<!ELEMENT val (#PCDATA)>
```

a) Donner un document contenant 5 éléments `tree` et valide pour la DTD.

b) On suppose que chaque élément `val` contient un entier. Écrire un schématron qui vérifie que la valeur d'un élément `val` est inférieure aux valeurs contenues dans chacun des éléments `val` enfants de ses deux frères `tree` quand ceux-ci existent.

► Exercice 3 1) Écrire une feuille de style XSLT qui prend un fichier XML quelconque et qui remplace chaque élément `x` par un élément `y` ayant les mêmes attributs avec les mêmes valeurs et qui laisse inchangé tous les autres éléments du documents.

2) Écrire une feuille de style XSLT qui prend un fichier XML quelconque et qui supprime tous les éléments dont le contenu textuel est vide.

► Exercice 4 On considère un document XML dans un fichier `exercice4.xml`, très inspiré des documents Gedcom, écrit ci-dessous sur deux colonnes.

<?xml version="1.0" encoding="utf-8"?>	<indi id="I5">
<gedcom>	<name>Caroline /Rabin/</name>
<fam id="F1">	<fams idref="F2"/>
<husb idref="I1"/>	</indi>
<wife idref="I2"/>	<indi id="I4">
<chil idref="I3"/>	<name>Julie /Turing/</name>
<chil idref="I4"/>	<fams idref="F3"/>
</fam>	</indi>
<fam id="F2">	<indi id="I6">
<husb idref="I3"/>	<name>Nicolas /Büchi/</name>
<wife idref="I5"/>	<fams idref="F3"/>
<chil idref="I7"/>	</indi>
<chil idref="I9"/>	<indi id="I7">
</fam>	<name>Arnaud /Turing/</name>
<fam id="F3">	<fams idref="F2"/>
<husb idref="I6"/>	<fams idref="F4"/>
<wife idref="I4"/>	</indi>
<chil idref="I7"/>	<indi id="I8">
<chil idref="I11"/>	<name>Laurence /Church/</name>
</fam>	<fams idref="F4"/>
<fam id="F4">	</indi>
<husb idref="I7"/>	<indi id="I9">
<wife idref="I8"/>	<name>Manuel /Turing/</name>
<chil idref="I14"/>	<fams idref="F2"/>
</fam>	<fams idref="F5"/>
<fam id="F5">	</indi>
<husb idref="I9"/>	<indi id="I10">
<wife idref="I10"/>	<name>Nicole /von Neumann/</name>
</fam>	<fams idref="F5"/>
<indi id="I1">	</indi>
<name>Jacques /Fraga/</name>	<indi id="I11">
<fams idref="F1"/>	<name>Dylan /Büchi/</name>
</indi>	<fams idref="F3"/>
<indi id="I2">	</indi>
<name>Estelle /Knuth/</name>	<indi id="I17">
<fams idref="F1"/>	<name>Sylvain /Büchi/</name>
</indi>	<fams idref="F3"/>
<indi id="I3">	</indi>
<name>Frederic /Turing/</name>	<indi id="I14">
<fams idref="F1"/>	<name>Françoise /Turing/</name>
<fams idref="F2"/>	<fams idref="F4"/>
</indi>	</indi>
	</gedcom>

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" encoding="utf-8"
    doctype-public="-//W3C//DTD XHTML 1.1//EN"
    doctype-system="http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"
    indent="yes"/>
  <xsl:key name="indfam" match="indifam" use="@id"/>
  <xsl:param name="idparam"/>
  <xsl:template match="/">
    <html>
      <head><title>Exercice 4</title></head>
      <body>
        <h1>Exercice 4</h1>
        <ul><xsl:apply-templates select="key('indfam', $idparam)"/></ul>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="indi">
    <li>
      <xsl:value-of select="name"/>
      <xsl:text> [</xsl:text><xsl:value-of select="@id"/><xsl:text>]</xsl:text>
      <xsl:if test="famc">
        <ul>
          <xsl:variable name="fam" select="key('indfam', famc/@idref)"/>
          <xsl:for-each select="$fam/husb | $fam/wife">
            <xsl:apply-templates select="key('indfam', @idref)"/>
          </xsl:for-each>
        </ul>
      </xsl:if>
    </li>
  </xsl:template>
</xsl:stylesheet>

```

- a) Donner le résultat de l'application de la feuille de style suivante avec la commande
 xsltproc --stringparam idparam I14 exercice4.xsl exercice4.xml
 où exercice4.xml et exercice4.xsl sont respectivement les noms des fichiers contenant le
 document XML précédent et la feuille de style précédente.

- b) Expliquer ce que fait la feuille de style précédente.